Pattern Recognition Letters 31 (2010) 2043–2051

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Qualitative test-cost sensitive classification

Mumin Cebe, Cigdem Gunduz-Demir *

Department of Computer Engineering, Bilkent University, Bilkent, Ankara 06800, Turkey

ARTICLE INFO

Article history:
Received 14 January 2009
Available online 1 June 2010
Communicated by R.P.W. Duin

Keywords:
Cost-sensitive learning
Qualitative decision theory
Feature extraction cost
Feature selection

ABSTRACT

This paper reports a new framework for test-cost sensitive classification. It introduces a new loss function definition, in which misclassification cost and cost of feature extraction are combined *qualitatively* and the loss is conditioned with current and estimated decisions as well as their *consistency*. This loss function definition is motivated with the following issues. First, for many applications, the relation between different types of costs can be expressed roughly and usually only in terms of ordinal relations, but not as a precise quantitative number. Second, the redundancy between features can be used to decrease the cost; it is possible not to consider a new feature if it is consistent with the existing ones. In this paper, we show the feasibility of the proposed framework for medical diagnosis problems. Our experiments demonstrate that this framework is efficient to significantly decrease feature extraction cost without decreasing accuracy.

1. Introduction

In the general framework of classification algorithms, cost of misclassification errors is typically considered for the design of classifiers (Duda et al., 2001; Turney, 2000). However, in many real-world applications, one may also want to balance misclassification cost with cost of feature extraction. For example, in medical diagnosis, it is possible to obtain a large group of features from various medical tests. On the other hand, a doctor orders only a subset of them considering the distinctive power of the features together with their costs. Typically, more expensive tests provide more distinctive features. Thus, the doctor first asks a patient simple questions to comprehend the current health status of the patient, and then, if only necessary, orders some tests (typically simpler and cheaper ones) based on the answers of the questions. If these tests are not adequate to make decision, the doctor orders more tests (most probably more complex and more expensive ones) based on both the answers and the previous test results.

In literature, only a few studies incorporate the cost of feature extraction into the design of their classification algorithms. A large group of them focus on constructing decision trees in a most accurately but, at the same time, a least costly way. These studies define their splitting criterion as a function of both the information gain of a feature and its extraction cost (Nunez, 1991; Tan, 1993). Alternatively, they use the sum of misclassification and test costs as a splitting criterion (Sheng and Ling, 2006; Yang et al., 2006). These studies use a greedy approach to construct their decision trees. To prevent the drawbacks of the greedy behavior, lookahead strategies (Norton, 1989) and hybrid genetic algorithms (Turney, 1995) are also proposed. The second group of studies sequentially select features based on expected utility. They follow a greedy approach such that, at each step, they select a feature, extraction of which introduces the maximum expected utility (Yang et al., 2006; Gunduz, 2001; Zhang and Ji, 2006). For a feature, utility is defined in terms of gain of using the feature and cost of its extraction. Yang et al. (2006) and Gunduz (2001) define the gain as the difference between the current misclassification cost and the one expected after feature extraction. These studies estimate the latter cost since it is not possible to know its exact value before extracting the feature. Yang et al. (2006) estimate it by taking expectation over all possible feature values. Gunduz (2001) first estimates the feature value by using the previously extracted features and then computes expected cost by employing the estimated feature as well as the previously extracted ones. Zhang and Ji (2006) define the gain as mutual information. They use dynamic Bayesian networks to estimate posteriors that are used in expected entropy computation. The third group of studies formulate the problem with a Markov decision process model. They first learn an optimal policy that minimizes the total expected cost on this model and then select features according to this policy. Zubek and Dietterich (2002) define a state for each possible combination of features and find the optimal policy via a non-greedy approach. Since such an approach requires high computational complexity, Ji and Carin (2007) propose an approximation to effectively find the optimal policy. This approximation uses a model, in which states are tied to mixture components of particular features and they are only partially observable.

* Corresponding author. Tel.: +90 312 290 3443; fax: +90 312 266 4047.
E-mail addresses: mumin@cs.bilkent.edu.tr (M. Cebe), gunduz@cs.bilkent.edu.tr (C. Gunduz-Demir).

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2010.05.028

Although these previous studies yield promising results, none of them addresses the following issues that are usually important for real-world applications. First, in these studies, misclassification cost and cost of feature extraction are combined quantitatively for the definition of a loss/utility function. For that, the misclassification cost is expressed as a precise quantitative value that is selected by considering the cost of feature extraction[1] and its importance over the misclassification cost. However, in many real-world applications, decision makers cannot express such importance in terms of precise quantitative values. Instead, they roughly express it in terms of ordinal relations; for instance, in cancer diagnosis, it can be expressed that the cost of a medical test is smaller than that of misdiagnosis. Second, all these studies select features based on current information and the one expected after feature extraction. None of them considers the consistency between this information. On the other hand, in many real-world applications, consistency is important. For example, in medical diagnosis, a doctor may not order an expensive test for a patient, if he/she is confident enough that the test confirms his/her current decision about the patient. Instead, the doctor may want to order a test, for which he/she thinks that it will change his/her decision. By doing so, the cost of extra tests, and hence, the overall cost can significantly be decreased without decreasing diagnosis accuracy.

In this paper, we report a novel test-cost sensitive approach that successfully addresses these issues. In our approach, we use a Bayesian decision theoretical framework, in which (1) misclassification cost and cost of feature extraction are combined *qualitatively* and (2) the loss function is conditioned with the decisions taken using current and estimated information as well as their *consistency*. In our previous study, we also consider the consistency by conditioning our loss function with the consistency between current and estimated decisions (Cebe and Gunduz-Demir, 2007). However, this previous study combines misclassification cost and cost of feature extraction quantitatively, which requires the user to determine exact quantitative constants. On the contrary, in this current work, we define the conditioned-loss function qualitatively, which does not require the user to express his/her prior information as precise quantitative numbers.

Qualitative decision theory studies the incorporation of qualitative knowledge into decision making problems (Doyle and Thomason, 1999). It enables to define probabilities and/or losses/utilities qualitatively, as opposed to the classical approach where these values should be defined as exact numerical values. This kind of qualitative definition allows the user to reflect his/her generic preferences on the problem, without the need of specifying them in terms of exact numerical values. There are many studies that focus on theoretical aspects of qualitative decision theory (Brafman and Tennenholtz, 1996; Dubois and Prade, 1995; Dubois et al., 2002; Fargier and Sabbadin, 2005; Lehmann, 2001; Pearl, 1993). However, its practical application is quite limited and there is still a large gap between the theory and the practice (Doyle and Thomason, 1999). The only application is the construction of qualitative probabilistic networks where the probabilistic relations between variables are defined by qualitative signs and inference is achieved by propagating the signs throughout the network (Brafman et al., 2004; Renooij and van der Gaag, 1998, 2002; Wellman, 1990). There are also studies that allow to represent uncertainties in misclassification costs. For instance, Adams and Hands (1999) define a comparative index of classifier performance when misclassification costs are not exactly known. However, these studies do not consider the problem of combining misclassification and feature extraction costs into a single loss/utility function for test-cost sensitive classification. In this work, we define qualitative conditioned-loss functions to reflect the generic preferences of the user on different types of costs and employ this representation for test-cost sensitive classification in medical diagnosis problems. Our experiments show that this qualitative representation significantly decreases the total test cost without decreasing diagnosis accuracy.

[1] Most of the time, feature extraction cost is easily expressed as a quantitative value. For example, in medical diagnosis, this cost can be expressed as the amount of money that one should pay for the corresponding medical test.

2. Methodology

In our approach, we define the loss function qualitatively and condition it with current and estimated decisions as well as their consistency. For a given instance x, the conditional risk $R(\alpha_i|x)$ of taking action α_i is

$$R(\alpha_i|x) = \sum_{j=1}^{N} P(C_j|x)\lambda(\alpha_i|C_j) \quad (1)$$

where $\{C_1,\ldots,C_N\}$ is the set of N possible classes, $P(C_j|x)$ is the probability of x belonging to class C_j, and $\lambda(\alpha_i|C_j)$ is the qualitative lost function for taking action α_i when the actual class is C_j. Comparing the conditional risks of all possible actions qualitatively, we take an action for which the conditional risk is qualitatively minimum. In this section, we first define our conditioned-loss function and derive conditional risk equations. Then, we incorporate *qualitativeness* into this loss function and explain how to *qualitatively* compare the conditional risks of actions. Finally, we provide the details of the proposed algorithm that uses this qualitative loss function.

2.1. Consistency-based loss functions

The proposed test-cost sensitive classification algorithm defines three types of actions: (1) $\texttt{extract}_k$ action that extracts feature F_k, (2) `classify` action that stops extraction and classifies the instance using current information, and (3) `reject` action that stops extraction and rejects the classification of the instance. Fig. 1 defines the loss function for each of these actions. The notations used in this figure as well as in the rest of the paper are summarized in Fig. 2.

For $\texttt{extract}_k$ action, the loss function always includes the extraction cost ($\texttt{cost}_k$) that should be paid for acquiring feature F_k. Additionally, it penalizes the extraction of F_k with a positive qualitative amount of `PENALTY` if the extraction does not yield correct classification ($C_k \neq C_{act}$). On the contrary, it rewards the extraction with a positive qualitative amount of `REWARD`, by adding $-$`REWARD` to the loss function, if the extraction yields correct classification by changing an incorrect current decision ($C_k = C_{act}$ but $C_{curr} \neq C_{act}$). However, it does not reward this action if the extraction just confirms a correct current decision ($C_k = C_{act}$ and $C_{curr} = C_{act}$) since this brings an additional cost without providing any new information. Therefore, the proposed loss function enforces

	$\lambda_{\texttt{extract}_k}$
$(C_k \neq C_{act})$	$\texttt{cost}_k + \texttt{PENALTY}$
$(C_k = C_{act})$ and $(C_{curr} = C_{act})$	$\texttt{cost}_k$
$(C_k = C_{act})$ and $(C_{curr} \neq C_{act})$	$\texttt{cost}_k - \texttt{REWARD}$
	$\lambda_{\texttt{classify}}$
$(C_{curr} = C_{act})$	$-\texttt{REWARD}$
$(C_{curr} \neq C_{act})$	PENALTY
	$\lambda_{\texttt{reject}}$
$(C_{curr} \neq C_{act})$ and $(C_k \neq C_{act})$, $\forall C_k \in C_{EST}$	$-\texttt{REWARD}$
$(C_{curr} = C_{act})$ or $(C_k = C_{act})$, $\exists C_k \in C_{EST}$	PENALTY

Fig. 1. Definition of the conditioned-loss function for $\texttt{extract}_k$, `classify`, and `reject` actions.

Class definitions	
$C_{\mathtt{act}}$	: The actual class that an instance belongs to
$C_{\mathtt{curr}}$	: The class given by the current classifier that uses only the extracted features
$C_{\mathtt{k}}$	: The class estimated by the classifier that uses the extracted features plus feature $F_{\mathtt{k}}$, which has not been extracted yet
$C_{\mathtt{EST}}$	: The set of classes $C_{\mathtt{k}}$, which are to be estimated for every non-extracted feature $F_{\mathtt{k}}$
Posteriors	
$P_{\mathtt{act}}(\mathtt{j}) \equiv P(C_{\mathtt{act}} = \mathtt{j}\|x)$	and $P_{\mathtt{act}}(\mathtt{j}') \equiv P(C_{\mathtt{act}} \neq \mathtt{j}\|x)$
$P_{\mathtt{curr}}(\mathtt{j}) \equiv P(C_{\mathtt{curr}} = \mathtt{j}\|x)$	and $P_{\mathtt{curr}}(\mathtt{j}') \equiv P(C_{\mathtt{curr}} \neq \mathtt{j}\|x)$
$P_{\mathtt{k}}(\mathtt{j}) \equiv P(C_{\mathtt{k}} = \mathtt{j}\|x)$	and $P_{\mathtt{k}}(\mathtt{j}') \equiv P(C_{\mathtt{k}} \neq \mathtt{j}\|x)$
Loss functions	
$\lambda_{\mathtt{extract_k}} \equiv \lambda(\mathtt{extract_k} \mid C_{\mathtt{curr}}, C_{\mathtt{EST}}, C_{\mathtt{act}})$	
$\lambda_{\mathtt{classify}} \equiv \lambda(\mathtt{classify} \mid C_{\mathtt{curr}}, C_{\mathtt{EST}}, C_{\mathtt{act}})$	
$\lambda_{\mathtt{reject}} \equiv \lambda(\mathtt{reject} \mid C_{\mathtt{curr}}, C_{\mathtt{EST}}, C_{\mathtt{act}})$	
Conditional risks	
$R_{\mathtt{extract_k}} \equiv R(\mathtt{extract_k} \mid x, C_{\mathtt{curr}}, C_{\mathtt{EST}})$	
$R_{\mathtt{classify}} \equiv R(\mathtt{classify} \mid x, C_{\mathtt{curr}}, C_{\mathtt{EST}})$	
$R_{\mathtt{reject}} \equiv R(\mathtt{reject} \mid x, C_{\mathtt{curr}}, C_{\mathtt{EST}})$	

Fig. 2. Notations used in the paper.

$$R_{\mathtt{extract_k}} = \sum_{j=1}^{N} P_{\mathtt{act}}(\mathtt{j})\lambda_{\mathtt{extract_k}} = \sum_{j=1}^{N} P_{\mathtt{act}}(\mathtt{j})\Big(\mathtt{cost_k} + P_{\mathtt{k}}(\mathtt{j}')\mathtt{PENALTY} + P_{\mathtt{k}}(\mathtt{j})P_{\mathtt{curr}}(\mathtt{j}')[-\mathtt{REWARD}]\Big) \quad (2)$$

$$R_{\mathtt{classify}} = \sum_{j=1}^{N} P_{\mathtt{act}}(\mathtt{j})\lambda_{\mathtt{classify}} = \sum_{j=1}^{N} P_{\mathtt{act}}(\mathtt{j})\Big(P_{\mathtt{curr}}(\mathtt{j})[-\mathtt{REWARD}] + P_{\mathtt{curr}}(\mathtt{j}')\mathtt{PENALTY}\Big) \quad (3)$$

$$R_{\mathtt{reject}} = \sum_{j=1}^{N} P_{\mathtt{act}}(\mathtt{j})\lambda_{\mathtt{reject}} = \sum_{j=1}^{N} P_{\mathtt{act}}(\mathtt{j})\Big(P_{\mathtt{curr}}(\mathtt{j}') \prod_{C_{\mathtt{k}} \in C_{\mathtt{EST}}} P_{\mathtt{k}}(\mathtt{j}')[-\mathtt{REWARD}] + \Big[1 - P_{\mathtt{curr}}(\mathtt{j}') \prod_{C_{\mathtt{k}} \in C_{\mathtt{EST}}} P_{\mathtt{k}}(\mathtt{j}')\Big]\mathtt{PENALTY}\Big) \quad (4)$$

the algorithm not to extract additional features when they are expected to confirm the correct current decision. This leads to less costly but equally accurate results. Here, we introduce the consistency mechanism, which plays an important role in reducing feature redundancy. It suggests extracting an additional feature only if the expected decision after using this feature is *inconsistent* with the incorrect current decision (i.e., if the feature is non-redundant). Otherwise, if the expected and current decisions are *consistent* (i.e., if the feature is redundant), it suggests not extracting the feature. The extraction is never rewarded if it is expected to give misclassification, regardless of whether it is consistent or inconsistent with the current decision.

For `classify` action, the loss function rewards the classification with `REWARD` if the current decision is correct ($C_{\mathtt{curr}} = C_{\mathtt{act}}$) and penalizes it with `PENALTY` otherwise ($C_{\mathtt{curr}} \neq C_{\mathtt{act}}$). Therefore, for correct current decisions, the loss function enforces the algorithm to classify the instance without extracting any additional feature.

For `reject` action, the loss function rewards the rejection of classification and feature extraction with `REWARD`, if both the current and estimated decisions yield misclassification ($C_{\mathtt{curr}} \neq C_{\mathtt{act}}$ and $C_{\mathtt{k}} \neq C_{\mathtt{act}}$ for every $C_{\mathtt{k}}$ in $C_{\mathtt{EST}}$). It penalizes the rejection with `PENALTY` if either the current decision or any of the estimated decisions yields the correct classification ($C_{\mathtt{curr}} = C_{\mathtt{act}}$ or $C_{\mathtt{k}} = C_{\mathtt{act}}$ for at least one $C_{\mathtt{k}}$ in $C_{\mathtt{EST}}$). Thus, the loss function enforces the algorithm to stop and reject classification when the correct classification is not possible. Here, `reject` action is important in reducing feature extraction cost as it causes to stop extracting new additional features if it is believed that no further feature would give correct classification.

Using this loss function, the conditional risks for $\mathtt{extract_k}$, `classify`, and `reject` actions are given in Eqs. (2)–(4). Our previous work defines the loss function and conditional risks similarly (Cebe and Gunduz-Demir, 2007). However, it requires using precise quantitative values of `REWARD` and `PENALTY`. In contrast, this current work defines `REWARD` and `PENALTY` as *qualitative values*, which eliminates the necessity of knowing their exact values to compute the conditional risks.

2.2. Qualitative decision making

Qualitative reasoning concerns with the development of methods that allow designing systems without precise quantitative information. It primarily uses ordinal relations between quantities, especially at particular locations ("landmark values"). The numerical value of a landmark may or may not be known, but the ordinal relations with respect to the landmark, reflecting the generic preferences, are known (Kuipers, 1994).

In this work, the landmarks are feature extraction costs ($\mathtt{cost_k}$) and `PENALTY` and `REWARD` values. Qualitative decision making requires qualitatively comparing conditional risks, in which these landmark values are used. Therefore, the ordering among the landmarks should be specified. In this paper, we focus on medical diagnosis problems and specify such an ordering for these problems making the following assumptions.

1. The cost of acquiring a feature (the price of a medical test) is expressed quantitatively and is exactly known. Thus, the extraction costs of different features are quantitatively compared among themselves.
2. `PENALTY` and `REWARD` are defined as positive numbers, but their exact values are not known. `PENALTY` is considered as the amount that should be paid for misdiagnosis and `REWARD` is considered as the amount that will be earned for correct diagnosis. It is assumed that `PENALTY` is always greater than `REWARD`. Thus, it has more tendency to preventing misdiagnosis. On the other hand, it is also possible to have the opposite assumption, where `REWARD` > `PENALTY`. In this case, the same method can be used to qualitatively compare conditional risks. However, the rules derived from these comparisons (the rules given by Cases 3 and 4 in Figs. 3–5) will be changed. The derivations of the new rules are given in Appendix A.
3. Feature extraction costs are always less than any partial amounts of `PENALTY` and `REWARD`. Thus, it is assumed that all tests are affordable to prevent misdiagnosis and lead to the correct one. This assumption results in neglecting $\mathtt{cost_k}$ against any amounts of `PENALTY` and `REWARD`. Its main motivation is the fact that for many real-world applications, misclassification

Case 1: $X_1 \geq 0, Y_1 \geq 0$		$\texttt{extract}_2$
Case 2: $X_1 < 0, Y_1 < 0$		$\texttt{extract}_1$
Case 3: $X_1 \geq 0, Y_1 < 0$	if $X_1 \geq \lvert Y_1 \rvert$	$\texttt{extract}_2$
	else if $X_1 / \lvert Y_1 \rvert \leq \texttt{SMALL}$	$\texttt{extract}_1$
	else	$\texttt{extract}_2$
Case 4: $X_1 < 0, Y_1 \geq 0$	if $\lvert X_1 \rvert \geq Y_1$	$\texttt{extract}_1$
	else if $\lvert X_1 \rvert / Y_1 \leq \texttt{SMALL}$	$\texttt{extract}_2$
	else	$\texttt{extract}_1$

Fig. 3. Decision rules derived for $\texttt{extract}_1$ vs. $\texttt{extract}_2$ comparison.

Case 1: $X_2 \geq 0, Y_2 \geq 0$		classify
Case 2: $X_2 < 0, Y_2 < 0$		$\texttt{extract}_k$
Case 3: $X_2 \geq 0, Y_2 < 0$	if $X_2 \geq \lvert Y_2 \rvert$	classify
	else if $X_2 / \lvert Y_2 \rvert \leq \texttt{SMALL}$	$\texttt{extract}_k$
	else	classify
Case 4: $X_2 < 0, Y_2 \geq 0$	if $\lvert X_2 \rvert \geq Y_2$	$\texttt{extract}_k$
	else if $\lvert X_2 \rvert / Y_2 \leq \texttt{SMALL}$	classify
	else	$\texttt{extract}_k$

Fig. 4. Decision rules derived for $\texttt{extract}_k$ vs. classify comparison.

Case 1: $X_3 \geq 0, Y_3 \geq 0$		reject
Case 2: $X_3 < 0, Y_3 < 0$		$\texttt{extract}_k$
Case 3: $X_3 \geq 0, Y_3 < 0$	if $X_3 \geq \lvert Y_3 \rvert$	reject
	else if $\frac{X_3}{\lvert Y_3 \rvert} \leq \texttt{SMALL}$	$\texttt{extract}_k$
	else	reject
Case 4: $X_3 < 0, Y_3 \geq 0$	if $\lvert X_3 \rvert \geq Y_3$	$\texttt{extract}_k$
	else if $\frac{\lvert X_3 \rvert}{Y_3} \leq \texttt{SMALL}$	reject
	else	$\texttt{extract}_k$

Fig. 5. Decision rules derived for $\texttt{extract}_k$ vs. reject comparison.

cost is commonly much greater than test costs and it is unrealistic to consider the quantitative values of these two types of cost in the same scale. For example, in cancer diagnosis, the cost of a medical test (e.g., an ultrasound scan) is much smaller than the misdiagnosis cost and obviously these costs are not in the same scale. Note that although we neglect their quantitative values, we consider the test costs through the consistency mechanism. That is, the proposed approach does not extract an additional feature if it is believed that the extraction just confirms current information.

Next subsections explain how to qualitatively compare actions pairwise using these assumptions and how to derive decision rules as a result of these comparisons.

2.2.1. $\texttt{extract}_1$ vs. $\texttt{extract}_2$

We compute the net risk to compare the conditional risks of $\texttt{extract}_1$ and $\texttt{extract}_2$ actions, which are defined for extracting features F_1 and F_2, respectively. Here we use Eq. (2) to express the conditional risks.

$$\begin{aligned} \mathit{NetRisk} &= R_{\texttt{extract}_1} - R_{\texttt{extract}_2} \\ &= (\texttt{cost}_1 - \texttt{cost}_2) + \sum_{j=1}^{N} P_{\texttt{act}}(\texttt{j}) \left(P_2(\texttt{j}) - P_1(\texttt{j}) \right) \texttt{PENALTY} \\ &\quad + \sum_{j=1}^{N} P_{\texttt{act}}(\texttt{j}) \left(P_2(\texttt{j}) - P_1(\texttt{j}) \right) P_{\texttt{curr}}(\texttt{j}') \texttt{REWARD} \\ &= \mathit{NetCost} + X_1 \texttt{PENALTY} + Y_1 \texttt{REWARD} \end{aligned} \tag{5}$$

where $\mathit{NetCost} = (\texttt{cost}_1 - \texttt{cost}_2)$, $X_1 = \sum_j P_{\texttt{act}}(\texttt{j})(P_2(\texttt{j}) - P_1(\texttt{j}))$, and $Y_1 = \sum_j P_{\texttt{act}}(\texttt{j})(P_2(\texttt{j}) - P_1(\texttt{j}))P_{\texttt{curr}}(\texttt{j}')$. Note that $P_{\texttt{act}}(\texttt{j})$ and $P_{\texttt{k}}(\texttt{j})$ are not known in advance, and hence, they should be estimated using current information beforehand. The details of this estimation are given in Section 2.3.1.

Negative values of *NetRisk* imply that the conditional risk of $\texttt{extract}_1$ is less than that of $\texttt{extract}_2$. Thus, $\texttt{extract}_1$ action is taken for negative *NetRisk*s and $\texttt{extract}_2$ action for nonnegative ones. The sign of *NetRisk* depends on the signs of X_1 and Y_1 since PENALTY and REWARD are defined as positive values and the sign of *NetCost* can be neglected because of the third assumption. Therefore, there are four different cases:

- *Case 1* ($X_1 \geq 0$ and $Y_1 \geq 0$).
  The values of both X_1PENALTY and Y_1REWARD are greater than or equal to zero, and hence, *NetRisk* is nonnegative. Therefore, $\texttt{extract}_2$ action is taken. If both $X_1 = 0$ and $Y_1 = 0$, the action with smaller cost is selected; the first assumption states that ordering among the test costs is known.
- *Case 2* ($X_1 < 0$ and $Y_1 < 0$).
  The values of X_1PENALTY and Y_1REWARD are less than zero, and hence, *NetRisk* is negative. Therefore, $\texttt{extract}_1$ action is taken.
- *Case 3* ($X_1 \geq 0$ and $Y_1 < 0$).
  The sign of *NetRisk* depends on the magnitudes of X_1 and Y_1. If $|X_1| \geq |Y_1|$ then $|X_1\texttt{PENALTY}| > |Y_1\texttt{REWARD}|$, as the second assumption states that PENALTY is greater than REWARD. Thus, *NetRisk* is nonnegative and $\texttt{extract}_2$ action is taken. If $|X_1| < |Y_1|$, we propose to use the definition given by Lehmann (2001) to compare $|X_1\texttt{PENALTY}|$ and $|Y_1\texttt{REWARD}|$.

Definition 1. Let A and B be positive. A is qualitatively greater than B if and only if there is a strictly positive real number $r \in (0,1)$ such that $(A - B)/A \geq r$.

Thus, $|Y_1\texttt{REWARD}|$ is qualitatively greater than $|X_1\texttt{PENALTY}|$ if and only if

$$\begin{aligned} &|Y_1\texttt{REWARD}| \succ |X_1\texttt{PENALTY}| \iff \frac{|Y_1\texttt{REWARD}| - |X_1\texttt{PENALTY}|}{|Y_1\texttt{REWARD}|} \geqslant r \\ &|Y_1\texttt{REWARD}| \succ |X_1\texttt{PENALTY}| \iff 1 - \frac{|X_1\texttt{PENALTY}|}{|Y_1\texttt{REWARD}|} \geqslant r \\ &|Y_1\texttt{REWARD}| \succ |X_1\texttt{PENALTY}| \iff \frac{|X_1|}{|Y_1|} \leqslant (1-r)\frac{\texttt{REWARD}}{\texttt{PENALTY}} \\ &|Y_1\texttt{REWARD}| \succ |X_1\texttt{PENALTY}| \iff \frac{|X_1|}{|Y_1|} \leqslant \texttt{SMALL} \end{aligned} \tag{6}$$

where $\texttt{SMALL} = (1 - r)(\texttt{REWARD}/\texttt{PENALTY})$ is a real number. This number is in between 0 and 1 as $r \in (0,1)$ and REWARD is assumed to be less than PENALTY, which implies $\texttt{REWARD}/\texttt{PENALTY} < 1$. Thus, if $|X_1| < |Y_1|$, we use Eq. (6) to determine the sign of *NetRisk*. If $|X_1/Y_1| \leqslant \texttt{SMALL}$ then $|Y_1\texttt{REWARD}|$ is qualitatively greater than $|X_1\texttt{PENALTY}|$, and hence, *NetRisk* is negative and $\texttt{extract}_1$ action is taken. Otherwise, if $|X_1/Y_1| > \texttt{SMALL}$,*NetRisk* is nonnegative and $\texttt{extract}_2$ action is taken.

Obviously, the selection of SMALL affects the decision. This work proposes to determine its value automatically from training data rather than having the user select this value. Thus, the selection does not require the user to express his/her belief in terms of quantitative numbers. Section 2.3.2 gives the details of this selection.

- *Case 4* ($X_1 < 0$ and $Y_1 \geq 0$).
  Likewise, the sign of *NetRisk* depends on the magnitudes of X_1 and Y_1. If $|X_1| \geq |Y_1|$ then $|X_1\texttt{PENALTY}| > |Y_1\texttt{REWARD}|$, since PENALTY is assumed to be greater than REWARD. Thus, *NetRisk* is negative and $\texttt{extract}_1$ action is taken. If $|X_1| < |Y_1|$, the values of $|X_1\texttt{PENALTY}|$ and $|Y_1\texttt{REWARD}|$ are qualitatively compared

using Eq. (6). In this case, if $|X_1/Y_1| \leqslant \texttt{SMALL}$, $|Y_1\texttt{REWARD}|$ is qualitatively greater than $|X_1\texttt{PENALTY}|$, and hence, *NetRisk* is nonnegative and $\texttt{extract}_2$ action is taken. Otherwise, if $|X_1/Y_1| > \texttt{SMALL}$, *NetRisk* is negative and $\texttt{extract}_1$ action is taken.

Fig. 3 provides a summary of these four different cases and lists the decision rules as a result of the comparisons.

2.2.2. $\texttt{extract}_k$ vs. classify

We compute the net risk using Eqs. (2) and (3) to compare the conditional risks of $\texttt{extract}_k$ and classify actions.

$$\begin{aligned} NetRisk &= R_{\texttt{extract}_k} - R_{\texttt{classify}} \\ &= \texttt{cost}_k + \sum_{j=1}^{N} P_{\texttt{act}}(\texttt{j}) \left(P_{\texttt{curr}}(\texttt{j}) - P_k(\texttt{j}) \right) \texttt{PENALTY} \\ &\quad + \sum_{j=1}^{N} P_{\texttt{act}}(\texttt{j}) \left(P_{\texttt{curr}}(\texttt{j}) - P_k(\texttt{j}) \right) P_{\texttt{curr}}(\texttt{j}')\texttt{REWARD} \\ &= \texttt{cost}_k + X_2\texttt{PENALTY} + Y_2\texttt{REWARD} \end{aligned} \tag{7}$$

where $X_2 = \sum_j P_{\texttt{act}}(\texttt{j})(P_{\texttt{curr}}(\texttt{j}) - P_k(\texttt{j}))$ and $Y_2 = \sum_j P_{\texttt{act}}(\texttt{j})(P_{\texttt{curr}}(\texttt{j}) - P_k(\texttt{j}))P_{\texttt{curr}}(\texttt{j}')$. The system takes $\texttt{extract}_k$ action if *NetRisk* is negative and classify action otherwise. Similarly, $\texttt{cost}_k$ term is neglected and there are four different cases depending on the signs of X_2 and Y_2. The decision rules are derived as explained in Section 2.2.1 and given in Fig. 4.

2.2.3. $\texttt{extract}_k$ vs. reject

We compute the net risk using Eqs. (2) and (4) to compare the conditional risks of $\texttt{extract}_k$ and reject actions.

$$\begin{aligned} NetRisk &= R_{\texttt{extract}_k} - R_{\texttt{reject}} = \texttt{cost}_k \\ &\quad + \sum_{j=1}^{N} P_{\texttt{act}}(\texttt{j}) \left(P_{\texttt{curr}}(\texttt{j}') \prod_{C_m \in C_{\texttt{REST}}} P_m(\texttt{j}') - P_k(\texttt{j}) \right) \texttt{PENALTY} \\ &\quad + \sum_{j=1}^{N} P_{\texttt{act}}(\texttt{j}) \left(P_{\texttt{curr}}(\texttt{j}') \prod_{C_m \in C_{\texttt{REST}}} P_m(\texttt{j}') - P_k(\texttt{j}) P_{\texttt{curr}}(\texttt{j}') \right) \texttt{REWARD} \\ &= \texttt{cost}_k + X_3\texttt{PENALTY} + Y_3\texttt{REWARD} \end{aligned} \tag{8}$$

where $X_3 = \sum_j P_{\texttt{act}}(\texttt{j})(P_{\texttt{curr}}(\texttt{j}') \prod P_m(\texttt{j}') - P_k(\texttt{j}))$ and $Y_3 = \sum_j P_{\texttt{act}}(\texttt{j})(P_{\texttt{curr}}(\texttt{j}') \prod P_m(\texttt{j}') - P_k(\texttt{j}) P_{\texttt{curr}}(\texttt{j}'))$. The system takes $\texttt{extract}_k$ action if *NetRisk* is negative and reject action otherwise. The decision rules are similarly derived, considering the signs of X_3 and Y_3, and given in Fig. 5.

2.2.4. classify vs. reject

We compute the net risk using Eqs. (3) and (4) to compare the conditional risks of classify and reject actions.

$$\begin{aligned} NetRisk &= R_{\texttt{reject}} - R_{\texttt{classify}} \\ &= \sum_{j=1}^{N} P_{\texttt{act}}(\texttt{j}) \left(P_{\texttt{curr}}(\texttt{j}) - P_{\texttt{curr}}(\texttt{j}') \prod_{C_m \in C_{\texttt{REST}}} P_m(\texttt{j}') \right) \texttt{PENALTY} \\ &\quad + \sum_{j=1}^{N} P_{\texttt{act}}(\texttt{j}) \left(P_{\texttt{curr}}(\texttt{j}) - P_{\texttt{curr}}(\texttt{j}') \prod_{C_m \in C_{\texttt{REST}}} P_m(\texttt{j}') \right) \texttt{REWARD} \\ &= X_4\texttt{PENALTY} + X_4\texttt{REWARD} \end{aligned} \tag{9}$$

where $X_4 = \sum_j P_{\texttt{act}}(\texttt{j})(P_{\texttt{curr}}(\texttt{j}) - P_{\texttt{curr}}(\texttt{j}') \prod P_m(\texttt{j}'))$. The system takes reject action if *NetRisk* is negative and classify action otherwise. In this comparison, we have the same multiplier for PENALTY and REWARD values. Thus, there are only two different cases depending on the multiplier sign. If $X_4 \geqslant 0$, *NetRisk* is nonnegative and classify action is taken. Otherwise, if $X_4 < 0$, *NetRisk* is negative and reject action is taken. The decision rules are given in Fig. 6.

2.3. Qualitative test-cost sensitive classification

For a given instance x, the proposed algorithm dynamically selects a subset of features for its classification. At a given time, it qualitatively compares the conditional risks of possible actions using the decision rules listed in Figs. 3–6 and selects the one with the minimum conditional risk. The algorithm continues this selection until either classify or reject action is taken. For the comparisons, X_i(X_1, X_2, X_3, and X_4), Y_i(Y_1, Y_2, and Y_3), and SMALL values should be estimated.

2.3.1. Posterior estimation

Posterior probability estimates are used to compute X_i and Y_i in Eqs. (5), (7)–(9). Posteriors $P_{\texttt{curr}}(\texttt{j})$ are computed by the current classifier using the features that have already been extracted. However, posteriors $P_k(\texttt{j})$ and $P_{\texttt{act}}(\texttt{j})$ cannot exactly be known prior to extracting feature F_k and they should be estimated using only the extracted features.

For each unextracted feature F_k, posteriors $P_k(\texttt{j})$ are estimated as follows: First, classifier $\mathscr{C}$ is trained on training samples $D = \{x_t\}_{t=1}^{T}$, for which the inputs include the extracted features plus feature F_k and the outputs are the class labels. Then, posteriors $P_k(\texttt{j})$ are generated for every training sample using classifier $\mathscr{C}$ and an estimator is trained to learn these generated posteriors from only the extracted features, but not feature F_k. The estimator is then used to estimate $P_k(\texttt{j})$ for unseen test instance x, without using its feature F_k. Note that for instance x, it is not possible to directly obtain $P_k(\texttt{j})$ using classifier $\mathscr{C}$ since its feature F_k has not been extracted yet.

In this work, we use a Parzen window estimator whose kernel function $\rho(u)$ defines a unit hypercube

$$\rho(u) = \begin{cases} 1 & \text{if } |u_i| \leqslant 1/2, \text{ for all dimensions } i \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Using this kernel function, posterior $P_k(\texttt{j})$ is estimated as

$$\widehat{P_k(\texttt{j})} = \frac{\sum_{t=1}^{T} \rho\left(\frac{x - x_t}{h}\right) \cdot P_{kt}(\texttt{j})}{\sum_{t=1}^{T} \rho\left(\frac{x - x_t}{h}\right)} \tag{11}$$

where h is the length of an edge of the hypercube and selected using leave-one out maximum likelihood estimation (Duin, 1976). In this equation, $P_k(\texttt{j})$ is equivalent to $P(C_k = \texttt{j}|x)$ as given in Fig. 2 and $P_{kt}(\texttt{j})$ is defined as $P(C_k = \texttt{j}|x_t)$.

For $\texttt{extract}_k$ action, posteriors $P_{\texttt{act}}(\texttt{j})$ are computed multiplying posteriors $P_{\texttt{curr}}(\texttt{j})$ and $P_k(\texttt{j})$ for each class j and normalizing the products such that $\sum_j P_{\texttt{act}}(\texttt{j}) = 1$. For classify and reject actions, only posteriors $P_{\texttt{curr}}(\texttt{j})$ are used since these actions stop further feature extractions for instance x.

Previous studies also estimate posteriors using the extracted features. Sheng and Ling (2006) and Yang et al. (2006) compute the posterior probability of a feature taking a particular value by using the Bayes' rule where likelihoods and priors are estimated by maximum likelihood estimation. Zhang and Ji (2006) compute posteriors using dynamic Bayesian networks. These studies conduct their experiments on discrete features. On the other hand, we work on both discrete and continuous features. In this work, we prefer using a non-parametric estimator, since our earlier

Case 1: $X_4 \geq 0$	classify
Case 2: $X_4 < 0$	reject

Fig. 6. Decision rules derived for classify vs. reject comparison.

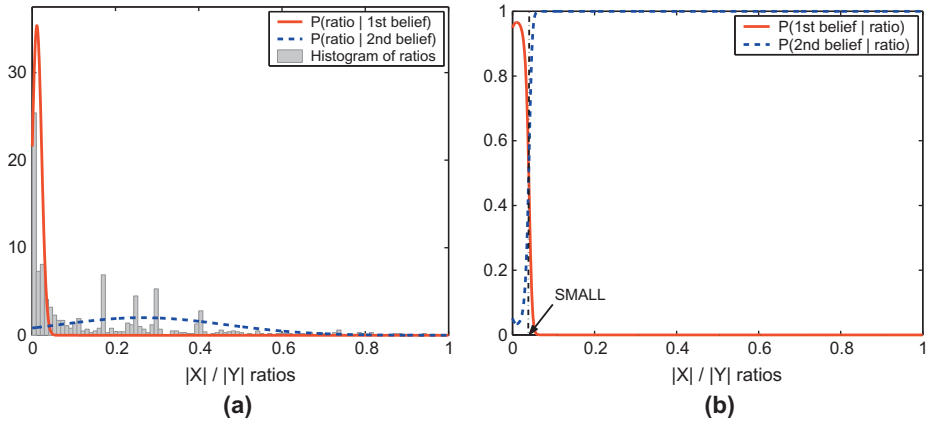

Fig. 7. Selection of SMALL value: (a) the histogram of the distinct $|X_i|/|Y_i|$ ratios of ambiguous cases and the two Gaussian components estimated on these ratios and (b) posteriors, which are obtained using the estimated Gaussians and prior probabilities.

experiments faced difficulties in selecting a parametric model that works with both discrete and continuous features as well as correctly estimating its parameters.

2.3.2. SMALL value estimation

The value of SMALL is automatically determined on the distinct training samples, for which the ambiguity arises (e.g., $|X_2| < |Y_2|$ in Case 3 of $\texttt{extract}_k$ vs. classify comparison). For these samples, we record $|X_i|/|Y_i|$ ratios and continue the algorithm by taking the SMALL value as zero; i.e., by quantitatively comparing $|X_i\texttt{PENALTY}|$ and $|Y_i\texttt{REWARD}|$. Such ambiguous cases are assumed to arise due to the possibility of two different beliefs (e.g., when $|X_2| < |Y_2|$ in Case 3 of $\texttt{extract}_k$-vs-classify comparison, one belief says to take $\texttt{extract}_k$ action whereas the other one says to take classify action). Thus, the $|X_i|/|Y_i|$ ratios of these ambiguous cases are assumed to be drawn from a mixture density of two Gaussian components,[2] each representing a different belief. These two Gaussian components and the priors of the two beliefs are estimated using an expectation–maximization algorithm. SMALL value is then determined as the point, where the posterior of the first belief is always smaller than that of the second one. For sample data, Fig. 7(a) shows the histogram of $|X_i|/|Y_i|$ ratios of ambiguous cases and the two Gaussian components estimated on these ratios. Fig. 7(b) shows posteriors of these beliefs.

3. Experiments

In our experiments, we use three medical data sets that are available in the UCI repository together with their costs (Blake and Merz, 1998). These data sets consist of features extracted by asking questions to a patient and those extracted from medical tests. A nominal cost of $1 is assigned to question-based features. Some medical tests may share a common cost (e.g., cost of collecting blood), which should be paid only once.

1. *Bupa liver disorders data set*: There are two classes and five *medical-test-based-features* with costs of {$7.27, $7.27, $7.27, $7.27, $9.86}. All medical tests share the common cost of $2.10. This data set includes 345 instances. As its size is relatively smaller, we use 10-fold cross-validation for this data set.

Table 1
Results obtained with decision tree classifiers.

	Baseline	Our algorithm		
	Accuracy	Accuracy	Cost red. percent	No. of rejects
Bupa	59.2 ± 5.3	59.0 ± 6.3	69.0 ± 7.2	0
Heart	77.1 ± 5.7	76.4 ± 6.9	63.1 ± 20.1	0
Thyroid	98.5	98.1	53.0	1

Table 2
Results obtained when consistency is not considered.

	Consistency-off		
	Accuracy	Cost red. percent	No. of rejects
Bupa	55.7 ± 9.0	24.5 ± 7.6	2
Heart	76.6 ± 6.1	1.7 ± 3.4	5
Thyroid	98.2	5.3	3

2. *Heart disease data set*: There are two classes and 13 features. Four of these features are *question-based-features* and the remaining nine are *medical-test-based-features* with costs of {$7.27,$5.20,$102.90,$102.90,$87.30,$87.30,$87.30, $15.50, $100.90}. There are three types of common costs: $2.10 for the first two features, $101.90 for the next two features, $86.30 for the next three features. The last two features do not share a common cost. This data set includes 303 instances. However, we eliminate six of them with missing values and use the remaining 297 instances. As its size is relatively smaller, we also use 10-fold cross-validation for this data set.
3. *Thyroid disease data set*: There are three classes and 21 features. The first 16 features are *question-based-features* and the next four are *medical-test-based-features* with costs of {$22.78,$11.41,$14.51,$11.41} and a common cost of $2.10. The last feature is defined using the nineteenth and the twentieth features, so we use it in classification only if both features have been extracted. This data set includes 3772 training instances. In the UCI repository, there is a separate test set including 3428 instances.

In our experiments, we use decision tree classifiers and Parzen window estimators.[3] Decision trees are trained using PRTools toolbox (Duin, 2000). Information gain is selected as the splitting crite-

[2] Here we use a Gaussian model, which is analytically tractable and often considered as an appropriate model for many real-world situations (Duda et al., 2001). However, it is also possible to use different models for SMALL selection. The investigation of such models could be considered as future work.

[3] This paper does not focus on increasing the absolute performance, but rather on demonstrating the methodology. However, the proposed methodology allows to use different classifiers that could yield better performances.

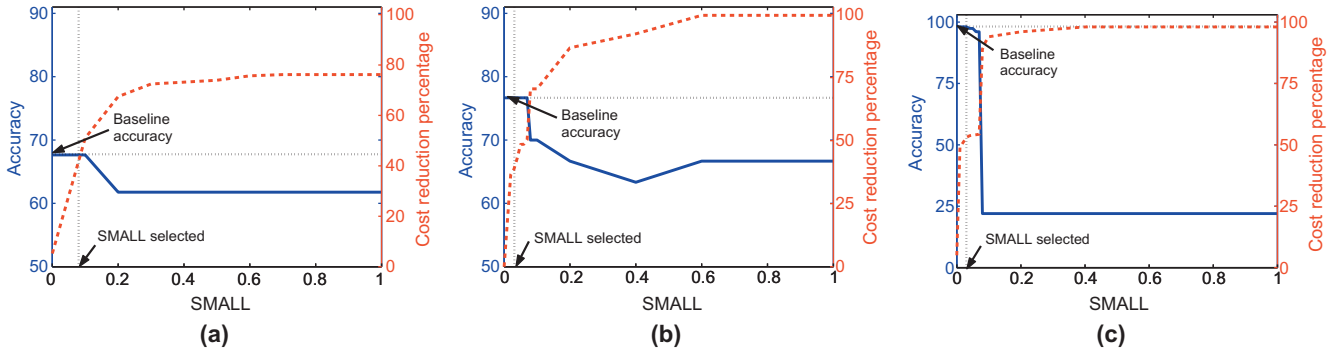

Fig. 8. Effects of the selection of SMALL to the accuracy and the cost reduction percentage. Results are obtained on the test set for: (a) the Bupa data set, (b) the Heart data set, and (c) Thyroid data set. The SMALL value selected on training samples and the accuracy of the baseline classifier are also indicated.

Table 3
Results obtained with HMM classifiers.

	Baseline	Our algorithm		(Ji and Carin, 2007)	
	Accuracy	Accuracy	Cost red. percent	Accuracy	Cost red. percent
Bupa	62.9 ± 7.2	62.0 ± 7.1	53.6 ± 14.5	61.8 ± 6.3	29.6 ± 8.3
Heart	85.9 ± 6.1	85.5 ± 5.9	37.0 ± 5.4	84.5 ± 6.0	35.1 ± 6.1
Thyroid	95.7	95.6	46.6	94.8	52.9

rion and early pruning option is used for the Bupa and Heart data sets and no pruning option is used for the Thyroid data set. The parameter of early pruning is selected as to optimize the baseline classifier. Other intermediary classifiers used for posterior estimation are trained using the selected parameter. Although this parameter may be non-optimal for all these classifiers, using the same parameter reduces time to search an optimal setup for each. Table 1 reports the results of the proposed qualitative test-cost sensitive algorithm and the baseline classifier, which uses all available features in its decision tree construction. This table provides accuracy, cost reduction percentage, and number of samples for which `reject` action is taken. The results are obtained on the test set for the Thyroid data set[4] and using 10-fold cross-validation for the Bupa and Heart data sets. For the Bupa and Heart data sets, the average accuracies and cost reduction percentages obtained with 10-fold cross-validation and their standard deviations are reported. These results demonstrate that the proposed qualitative test-cost sensitive algorithm significantly decreases overall feature extraction cost without decreasing accuracy. The results also show that `reject` action is only rarely taken.

Our algorithm starts with the cheapest feature and sequentially selects a subset of other features until `classify` or `reject` action is taken. In order to analyze the effects of starting with a more expensive feature, we repeat the experiments for the Bupa data set starting with the most distinctive but more expensive feature. Ten-fold cross-validation gives 58.5% accuracy and 43.9 cost reduction percentage. Although the accuracy is almost the same with the accuracy given in Table 1, there is an approximately 20 percent decrease in the cost reduction. This decrease is attributed to the fact that there is typically a direct proportion between the cost of features and their distinctive powers. When the algorithm starts with a more distinctive feature, it should pay its cost for any instance regardless of whether this feature is actually necessary for the instance.

In order to examine its importance, we repeat the experiments without using consistency. For that, we always reward feature extraction if it yields correct classification, regardless of whether or not this classification would be consistent with the current decision. Table 2 gives the results. They show that the algorithm tends to extract almost all of the features when it does not employ consistency. This is presumably due to the assumption of misclassification cost being greater than extraction cost of any feature. On the other hand, with the use of consistency, our algorithm can stop extracting features if it believes that future decisions are to be consistent with the current one. This prevents extracting redundant features.

We also investigate the effects of SMALL selection to the results. Fig. 8 gives accuracies (with solid blue curves and using the left *y*-axis) and cost reduction percentages (with dashed red curves and using the right y-axis) as a function of SMALL. It shows the test results for the Thyroid data set and the results of a single fold for the Bupa and Heart data sets. It also gives the selected SMALL value and accuracies of baseline classifiers (with dotted black curves and using the left *y*-axis). For the Bupa and Heart data sets, the accuracy change with respect to SMALL is relatively smaller whereas the change in the cost reduction is larger. This shows that the algorithm attempts to yield higher accuracies at the cost of decreasing cost reduction. For the Thyroid data set, SMALL affects both accuracy and cost reduction. Smaller values should be used to obtain higher accuracies; the algorithm can successfully select one of such values.

3.1. Comparisons

We compare our results with those of the previous algorithm,[5] which employs a partially observable Markov decision process (POMDP) to solve the feature selection problem (Ji and Carin, 2007). This previous work uses an extension of a standard hidden Markov model (HMM) classifier where state transition probabilities are conditioned with feature extraction actions and values observed after feature extraction. This model can be used in two

[4] Our previous work (Cebe and Gunduz-Demir, 2007) takes the cost of *question-based-features* as zero (instead of a nominal cost of $1) and does not consider the common costs. Thus, its results for the Thyroid data set are slightly different than those in given Table 1.

[5] We would like to thank the authors for kindly sharing their code with us.

Case 1: $X_1 + Y_1 \geq 0$	$\texttt{extract}_2$
Case 2: $X_1 + Y_1 < 0$	$\texttt{extract}_1$

Fig. 9. Decision rules derived for $\texttt{extract}_1$ vs. $\texttt{extract}_2$ comparison when PENALTY = REWARD.

different ways: (1) When a feature sequence is specified, it takes actions depending on the sequence and produces the probability of the sequence being generated by the model of each class (i.e., class posterior probabilities). (2) When a feature sequence is not specified, it sequentially determines a sequence of features, calculating expected risk of extracting each remaining feature with the POMDP and using expected risk of taking classify action. It considers the remaining features extraction of which decreases the risk of classify action by at least an amount of their extraction costs and selects the one with the maximum net decrease. If there are no such remaining features, the algorithm stops and classifies the sample using the extracted features.

The proposed algorithm and the baseline classifier use the HMM as described in the first way. The proposed algorithm obtains posteriors providing a feature subset to the HMM whereas the baseline classifier obtains them providing the complete set of features. The HMM model has a parameter (the number of states); this parameter is also selected as to optimize the baseline classifier. Table 3 reports the results of the proposed algorithm, the previous algorithm (Ji and Carin, 2007), and the baseline classifier. Although all of them use the same HMM, the baseline classifier employs all features whereas the others have their own feature selection policies. The policy of Ji and Carin (2007) has two free model parameters: cost of correct classification and cost of misclassification. These parameters are selected on training samples. On the other hand, the feature selection policy of the proposed algorithm does not require any free model parameter being externally optimized; there is no need for the user to determine the value of SMALL beforehand since it is automatically determined on training samples.

In order to statistically analyze the results given in Table 3, we conduct statistical tests. The Wilcoxon signed rank test is used for cost reduction percentages and the McNemar's test is used for accuracies. Both tests use a significance level of 0.05.

For the Bupa data set, there exists no statistically significant difference between accuracies. However, the difference between cost reductions is statistically significant. This difference is related with features selected by the algorithms. The proposed algorithm usually stops after selecting a single feature as it believes that no additional feature will change its decision. This indicates the importance of consistency. On the other hand, the previous algorithm (Ji and Carin, 2007) continues extracting additional features. This algorithm proposes a myopic approach to approximate the non-myopic POMDP solution. As indicated by its authors, such an approximation may not be effective for some examples and the Bupa data set may be one of them. For the Heart data set, there exists no statistically significant difference between accuracies and cost reductions. For the Thyroid data set, the proposed algorithm yields statistically better accuracies whereas the previous algorithm leads to statistically better cost reductions. Here the baseline HMM classifier gives more inaccurate results (more inaccurate posteriors) compared to decision trees. This causes the proposed algorithm to take incorrect decisions in feature selection; it attempts to improve accuracy at the cost of extracting more and more features since misclassification cost is assumed to be always greater than the extraction cost of any features.

In these results, the proposed algorithm does not take reject action for the Bupa and Heart data sets and it takes reject action for less than one percent of the instances in the Thyroid data set.

Case 1: $X_1 \geq 0, Y_1 \geq 0$		$\texttt{extract}_2$
Case 2: $X_1 < 0, Y_1 < 0$		$\texttt{extract}_1$
Case 3: $X_1 \geq 0, Y_1 < 0$	if $\lvert Y_1 \rvert \geq X_1$	$\texttt{extract}_1$
	else if $\lvert Y_1 \rvert / X_1 \leq$ SMALL2	$\texttt{extract}_2$
	else	$\texttt{extract}_1$
Case 4: $X_1 < 0, Y_1 \geq 0$	if $Y_1 \geq \lvert X_1 \rvert$	$\texttt{extract}_2$
	else if $Y_1 / \lvert X_1 \rvert \leq$ SMALL2	$\texttt{extract}_1$
	else	$\texttt{extract}_2$

Fig. 10. Decision rules derived for $\texttt{extract}_1$ vs. $\texttt{extract}_2$ comparison when PENALTY < REWARD.

This is presumably due to the inaccurate posteriors generated by the HMM classifier. Note that in computing accuracies and in conducting statistical tests, we consider the reject cases as incorrect classifications. Table 3 also shows that the proposed algorithm can use any type of classifiers since it uses posteriors regardless of the classifier type. When the results in Table 1 (a decision tree classifier) and Table 3 (an HMM classifier) are compared, it can be seen that the accuracy of our algorithm depends on the accuracy of the baseline classifier.

4. Conclusion

We introduced a new Bayesian decision theoretical framework for test-cost sensitive classification. This framework uses a new loss function in which misclassification cost and cost of feature extraction are qualitatively combined and the loss function is conditioned with current and estimated decisions as well as their consistency. Working with three medical diagnosis problems, our experiments demonstrated that (1) the proposed approach significantly decreases overall feature extraction cost without decreasing diagnosis accuracy, and (2) it overcomes the problem for the user to express his/her prior belief (the relation between misclassification cost and cost of feature extraction) as an exact quantitative number.

One of the future research directions is to investigate incorporation of the qualitative decision theory into other machine learning problems. Another possibility is to also include the other types of cost (e.g., delay cost (Sheng and Ling, 2006) and computational cost (Demir and Alpaydin, 2005)) into the problem formulation.

Appendix A

This work assumes that PENALTY > REWARD. However, it is also possible to have other assumptions (PENALTY = REWARD or REWARD > PENALTY), for which conditional risks can qualitatively be compared using the same method explained in Section 2.2. Although the method is the same, the rules given in Figs. 3–5 are partially changed. This appendix derives the rules of $\texttt{extract}_1$ vs. $\texttt{extract}_2$ comparison for the other assumptions (Figs. 9 and 10). It uses the same *NetRisk* equation, given in Eq. (5), and takes $\texttt{extract}_1$ action for negative values of *NetRisk* and $\texttt{extract}_2$ action for nonnegative ones. It also neglects *NetCost* against any partial amount of PENALTY and REWARD.

When PENALTY = REWARD, Eq. (5) becomes *NetRisk* = *NetCost* + $(X_1 + Y_1)$ PENALTY. Since *NetCost* is neglected and PENALTY is always greater than zero, the sign of *NetRisk* depends on the sign of $(X_1 + Y_1)$. Thus, $\texttt{extract}_1$ action is taken for negative sums and $\texttt{extract}_2$ action for nonnegative ones.

When REWARD > PENALTY, the same four different cases are considered, depending on the sign of X_1 and Y_1. The decision rules for Cases 1 and 2 remain exactly the same. On the other hand, the

rules for Cases 3 and 4, where X_1 and Y_1 have the opposite signs, are to be changed.

- *Case 3* ($X_1 \geqslant 0$ and $Y_1 < 0$).
  The sign of *NetRisk* depends on the magnitudes of X_1 and Y_1. If $|Y_1| \geqslant |X_1|$ then $|Y_1\texttt{REWARD}| > |X_1\texttt{PENALTY}|$ since REWARD > PENALTY. Thus, *NetRisk* is negative and $\texttt{extract}_1$ action is taken. If $|Y_1| < |X_1|$, Definition 1 is used for qualitative comparison. $|X_1\texttt{PENALTY}|$ is qualitatively greater than $|Y_1\texttt{REWARD}|$ if and only if

$$|X_1\texttt{PENALTY}| \succ |Y_1\texttt{REWARD}| \iff \frac{|X_1\texttt{PENALTY}| - |Y_1\texttt{REWARD}|}{|X_1\texttt{PENALTY}|} \geqslant r_2$$

$$|X_1\texttt{PENALTY}| \succ |Y_1\texttt{REWARD}| \iff \frac{|Y_1|}{|X_1|} \leqslant (1 - r_2)\frac{\texttt{PENALTY}}{\texttt{REWARD}}$$

$$|X_1\texttt{PENALTY}| \succ |Y_1\texttt{REWARD}| \iff \frac{|Y_1|}{|X_1|} \leqslant \texttt{SMALL2} \qquad (12)$$

  where $\texttt{SMALL2} = (1 - r_2)(\texttt{PENALTY}/\texttt{REWARD})$ is a real number in between 0 and 1, as $r_2 \in (0,1)$ and PENALTY < REWARD. Thus, if $|Y_1| < |X_1|$, Eq. (12) is used to determine the sign of *NetRisk*. If $|Y_1|/|X_1| \leqslant \texttt{SMALL2}$, *NetRisk* is nonnegative and $\texttt{extract}_2$ action is taken. Otherwise, *NetRisk* is negative and $\texttt{extract}_1$ action is taken.
- *Case 4* ($X_1 < 0$ and $Y_1 \geqslant 0$).
  The sign of *NetRisk* depends on the magnitudes of X_1 and Y_1. If $|Y_1| \geqslant |X_1|$ then $|Y_1\texttt{REWARD}| > |X_1\texttt{PENALTY}|$, since REWARD > PENALTY. Thus, *NetRisk* is nonnegative and $\texttt{extract}_2$ action is taken. If $|Y_1| < |X_1|$, $|X_1\texttt{PENALTY}|$ and $|Y_1\texttt{REWARD}|$ are qualitatively compared using Eq. (12). In this case, if $|Y_1|/|X_1| \leqslant \texttt{SMALL2}$ then $|X_1\texttt{PENALTY}|$ is qualitatively greater than $|Y_1\texttt{REWARD}|$, and hence, *NetRisk* is negative and $\texttt{extract}_1$ action is taken. Otherwise, *NetRisk* is nonnegative and $\texttt{extract}_2$ action is taken.

References

Adams, N.M., Hands, D.J., 1999. Comparing classifiers when the misallocation costs are uncertain. Pattern Recognition 32, 1139–1147.

Blake, C.L., Merz, C.J., 1998. UCI Repository of Machine Learning Databases. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Brafman, R.I., Tennenholtz, M., 1996. On the foundations of qualitative decision theory. In: AAAI 1996, Portland, OR.

Brafman, R.I., Domshlak, C., Shimony, S.E., 2004. Qualitative decision making in adaptive presentation of structured information. ACM Trans. Inform. Syst. 22 (4), 503–539.

Cebe, M., Gunduz-Demir, C., 2007. Test-cost sensitive classification based on conditioned loss functions. In: ECML 2007, Warsaw, Poland.

Demir, C., Alpaydin, E., 2005. Cost-conscious classifier ensembles. Pattern Recognition Lett. 26 (14), 2206–2214.

Doyle, J., Thomason, R.H., 1999. Background to qualitative decision theory. AI Mag. 20 (2), 55–68.

Dubois, D., Prade, H., 1995. Possibility theory as a basis for qualitative decision theory. In: IJCAI 1995, San Francisco, CA.

Dubois, D., Fargier, H., Prade, H., Perny, P., 2002. Qualitative decision theory: from savage's axioms to nonmonotonic reasoning. J. ACM 49 (4), 455–495.

Duda, O.R., Hart, E.P., Stork, G.D., 2001. Pattern Classification. Wiley Interscience, New York.

Duin, R.P.W., 1976. On the choice of smoothing parameters for Parzen estimators of probability density functions. IEEE Trans. Comput. 25, 1175–1179.

Duin, R.P.W., 2000. PRTools 3.0, A Matlab Toolbox for Pattern Recognition. Delft University of Technology.

Fargier, H., Sabbadin, R., 2005. Qualitative decision under uncertainty: back to expected utility. Artif. Intell. 164, 245–280.

Gunduz, C., 2001. Value of Representation in Pattern Recognition. M.S. Thesis, Bogazici University, Istanbul, Turkey.

Ji, S., Carin, L., 2007. Cost-sensitive feature acquisition and classification. Pattern Recognition 40, 1474–1485.

Kuipers, B., 1994. Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge. MIT, Cambridge.

Lehmann, D., 2001. Expected qualitative utility maximization. Game Econ. Behav. 35 (12), 54–79.

Norton, S.W., 1989. Generating better decision trees. In: IJCAI 1989, Detroit, MI.

Nunez, M., 1991. The use of background knowledge in decision tree induction. Mach. Learn. 6, 231–250.

Pearl, J., 1993. From qualitative utility to conditional ought to. In: UAI 1993, San Mateo, CA.

Renooij, S., van der Gaag, L.C., 1998. Decision making in qualitative influence diagrams. In: FLAIRS Conference 1998, Menlo Park, CA.

Renooij, S., van der Gaag, L.C., 2002. From qualitative to quantitative probabilistic networks. In: UAI 2002, San Francisco, CA.

Sheng V.S., Ling, C.X., 2006. Feature value acquisition in testing: A sequential batch test. In: ICML 2006, New York, NY.

Tan, M., 1993. Cost-sensitive learning of classification knowledge and its applications in robotics. Mach. Learn. 13, 7–33.

Turney, P.D., 1995. Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm. J. Artif. Intell. Res. 2, 369–409.

Turney, P.D., 2000. Types of cost in inductive concept learning. In: Workshop on Cost-Sensitive Learning, ICML 2000, Stanford, CA.

Wellman, M.P., 1990. Fundamental concepts of qualitative probabilistic networks. Artif. Intell. 44 (3), 257–303.

Yang, Q., Ling, C., Chai, X., Pan, R., 2006. Test-cost sensitive classification on data missing values. IEEE Trans. Knowl. Data Eng. 18, 626–638.

Zhang, Y., Ji, Q., 2006. Active and dynamic information fusion for multisensor systems with dynamic Bayesian networks. IEEE Trans. Systems Man Cybernet. B 36.

Zubek, V.B., Dietterich, T.G., 2002. Pruning improves heuristic search for cost-sensitive learning. In: ICML 2002, San Francisco, CA.